



NVIDIA DOCA Secure Channel

Application Guide

Table of Contents

Chapter 1. Introduction.....	1
Chapter 2. System Design.....	2
Chapter 3. Application Architecture.....	4
Chapter 4. DOCA Libraries.....	5
Chapter 5. Configuration Flow.....	6
Chapter 6. Dependencies.....	7
Chapter 7. Running the Application.....	8
Chapter 8. Arg Parser DOCA Flags.....	10
Chapter 9. References.....	12

Chapter 1. Introduction

DOCA Comm Channel is a secure, network independent communication channel between the host and the NVIDIA® BlueField® DPU.

Comm channel allows the host to control services on the DPU, activate certain offloads, or exchange messages using client-server framework. Communication is based on RDMA queue-pairs (QPs). Each packet that is sent by the QPs consists of two parts: Header and data. The header is a 32-bit structure that holds metadata on the msg to allow the use of a handshake protocol, credit incremental over data packets, and error handling.

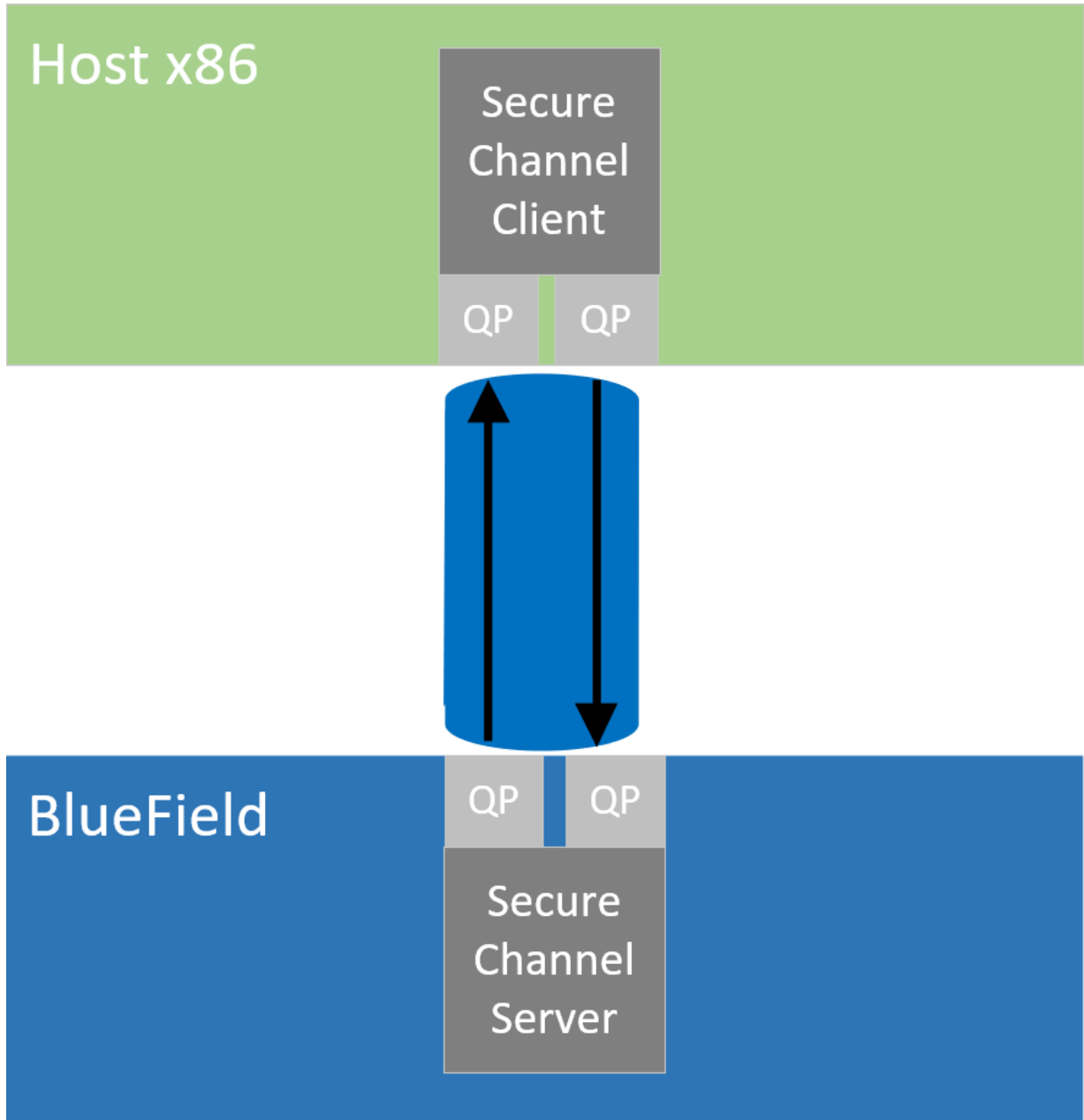
The client (host) side is able to communicate only with one server at a time while the server side is able to communicate with multiple clients.

The API allows communication between any PF/VF/SF on the host to the service on the Arm.

Secure channel allows the user to select the message size and amount to be exchanged between the client and the server to simulate heavy load on the channel.

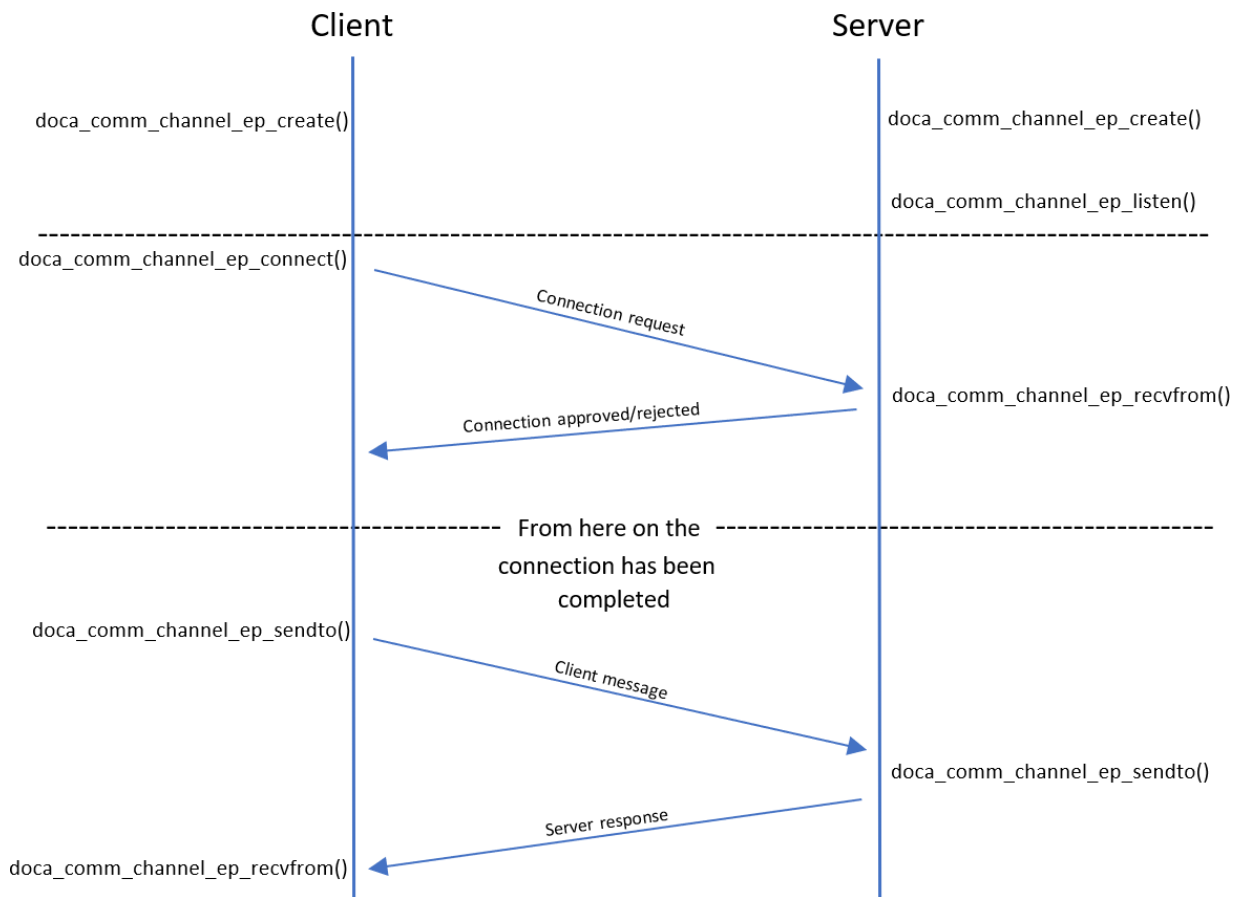
Chapter 2. System Design

A secure channel application runs on client mode (host) and server mode (DPU). Once a channel is open, messages can flow from both sides. Each endpoint holds 2 RDMA QPs, one for send operations and one for receive operations.



Chapter 3. Application Architecture

The secure channel application runs on top of the DOCA Comm Channel API. Full connection flow between the client and the server is illustrated in the following:



1. Both sides initiate `create()`.
2. Server listens and waits for new connections.
3. Server initiates `recvfrom()` to indicate it is ready to exchange messages.
4. Client executes `connect()` to server and starts connection initialization.
5. Client sends first message to server.
6. Server sends a response.

Chapter 4. DOCA Libraries

This application leverages the [DOCA Comm Channel library](#).

Chapter 5. Configuration Flow

1. Parse application argument.

- a). Initialize the arg parser resources and register DOCA general parameters.

```
doca_argp_init();
```

- b). Register application parameters.

```
register_secure_channel_params();
```

- c). Parse application flags.

```
doca_argp_start();
```

2. Run main logic.

```
sc_start();
```

- a). Initiate synchronization mechanism between send and receive threads.
- b). Initiate Comm Channel endpoint.
- c). Server side starts listening for new connections and client side connects to server.
- d). Initiate signal masking and epoll file descriptor.
- e). Start send and receive threads. Both threads share the same Comm Channel so each one must "lock" the channel before any send/receive operation.
- f). Send thread prints total number of messages successfully sent.
- g). Once Ctrl+C is entered in the shell, receive thread prints the total number of messages successfully received.
- h). Close and destroy resources.

Chapter 6. Dependencies

BlueField-2 firmware version 24.35.1012 or later.

Chapter 7. Running the Application

1. Refer to the following documents:

- ▶ [NVIDIA DOCA Installation Guide for Linux](#) for details on how to install BlueField-related software.
- ▶ [NVIDIA DOCA Troubleshooting Guide](#) for any issue you may encounter with the installation, compilation, or execution of DOCA applications.
- ▶ [NVIDIA DOCA Applications Overview](#) for additional compilation instructions and development tips for the DOCA applications.

2. The URL filtering example binary is located under `/opt/mellanox/doca/applications/secure_channel/bin/doca_secure_channel`. To build all the applications together, run:

```
cd /opt/mellanox/doca/applications/  
meson build  
ninja -C build
```

3. To build only the secure channel application:

a). Edit the following flags in `/opt/mellanox/doca/applications/meson_options.txt`:

- ▶ Set `enable_all_applications` to `false`
- ▶ Set `enable_secure_channel` to `true`

b). Run the commands in step 2.



Note: `doca_secure_channel` is created under `./build/secure_channel/src/`.

Application usage:

```
Usage: doca_secure_channel [DOCA Flags] [Program Flags]
```

DOCA Flags:

```
-h, --help           Print a help synopsis  
-v, --version        Print program version information  
-l, --log-level      Set the log level for the program  
<CRITICAL=20, ERROR=30, WARNING=40, INFO=50, DEBUG=60>
```

Program Flags:

```
-s, --msg-size       Message size to be sent  
-n, --num-msgs      Number of messages to be sent  
-p, --pci-addr      DOCA Comm Channel device PCI address
```

```
-r, --rep-pci          DOCA Comm Channel device representor PCI
address (needed only on DPU)
```



Note: For additional information on the app, use `-h`:

```
/opt/mellanox/doca/applications/secure_channel/bin/doca_secure_channel -h
```

4. CLI example for running the app on BlueField:

```
/opt/mellanox/doca/applications/secure_channel/bin/doca_secure_channel -s 256 -n
10 -p 03:00.0 -r b1:00.0
```

5. CLI example for running the app on the host:

```
/opt/mellanox/doca/applications/secure_channel/bin/doca_secure_channel -s 1024 -n
100 -p b1:00.0
```



Note: Refer to section "Running DOCA Application on Host" in [NVIDIA DOCA Virtual Functions User Guide](#).

6. To run `doca_secure_channel` using a JSON file:

```
doca_secure_channel --json [json_file]
```




For example:

```
cd /opt/mellanox/doca/applications/secure_channel/bin
./doca_secure_channel --json sc_server_params.json
```

Chapter 8. Arg Parser DOCA Flags

Refer to [NVIDIA DOCA Arg Parser Programming Guide](#) for more information.

Flag Type	Short Flag	Long Flag/ JSON Key	Description	JSON Content
General Flags	l	log-level	Sets the log level for the application: <ul style="list-style-type: none"> ▶ CRITICAL=20 ▶ ERROR=30 ▶ WARNING=40 ▶ INFO=50 ▶ DEBUG=60 	<code>"log-level": 60</code>
	v	version	Print program version information	N/A
	h	help	Print a help synopsis	N/A
Program Flags	s	msg-size	Message size in bytes <div style="border: 1px solid gray; background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  Note: This is a mandatory flag. </div>	<code>"msg-size": 128</code>
	n	num-msgs	Number of messages to send on both sides <div style="border: 1px solid gray; background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  Note: This is a </div>	<code>"num-msgs": 256</code>

Flag Type	Short Flag	Long Flag/ JSON Key	Description	JSON Content
			 mandatory flag.	
	p	pci-addr	DOCA Comm Channel device PCIe address  Note: This is a mandatory flag.	<code>"pci-addr": 03:00.1</code>
	r	rep-pci	DOCA Comm Channel device representor PCIe address  Note: This is a mandatory flag.	<code>"rep-pci": b1:00.1</code>

Chapter 9. References

- ▶ `/opt/mellanox/doca/applications/secure_channel/src`

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2023 NVIDIA Corporation & affiliates. All rights reserved.